Lightweight Fine-Tuning of LLMs for Explainable Intrusion Detection in SDN

Suvajit Lodh*, Islam Obaidat†, Furqan Rustam*‡, Anca Delia Jurcut‡
*School of Computing, National College of Ireland, Dublin, (Ireland),
†Department of CST, North Carolina A&T State University (USA)
‡School of Computer Science, University College Dublin (Ireland)
reachshuvojit@gmail.com, iaobaidat@ncat.edu, furqan.rustam@ucdconnect.ie, anca.jurcut@ucd.ie

***Abstract*—Cybersecurity concerns are rising with the rapid adoption of technology as cybercriminals grow more active. Protecting complex networks like Software-Defined Networking (SDN) is increasingly challenging because its centralized architecture introduces vulnerabilities that traditional security systems struggle to handle. This paper investigates the application of large language models (LLMs) for intrusion detection in SDN environments. Our proposed approach fine-tunes three LLMs, GPT_NEO, Phi-2, and Llama2-7b, through Quantized Low-Rank Adaptation (QLoRA), enabling efficient 4-bit quantization and reduced memory usage. Structured network features are transformed into natural language prompts for binary classification of benign and malicious traffic. Experimental results show that all models achieve high accuracy, with Llama2-7b and Phi-2 reaching high scores across multiple data scales. CodeCarbon tracking highlights the environmental trade-offs, with Llama2-7b consuming the most energy and Phi-2 being the most efficient. Our proposed framework for Phi-2 and GPT_NEO achieves a 1.00 accuracy score with the lowest CO_2 emission of 0.173 when compared with the baselines. Further, we explore explainability challenges for our LLM models, noting the limitations of token-level interpretability tools in handling dense textual embeddings.**

***Index Terms*—Intrusion Detection, LLM, Fine-tuning, Explainable AI, SDN**

I. Introduction

In today's interconnected world, network security remains a critical priority, as timely intrusion detection is essential to maintaining reliable communication systems [1]. With modern architectures, such as Software-Defined Networking (SDN), enabling centralized and programmable control, networks gain flexibility at the cost of new security challenges, making effective intrusion detection especially critical [2]. In addition, many AI-driven intrusion detection systems (including intrusion detection in SDN) often function as "black-box" models with limited transparency (a drawback that undermines operator trust and complicates incident response), highlighting explainable artificial intelligence (XAI) techniques crucial for providing interpretable insights and improving confidence in network defenses [3]. Large Language Models (LLMs) demonstrate strong generalization capabilities and the ability to learn complex patterns, making them promising tools for improving intrusion detection in SDN environments [4], [5]. Integrating LLMs into intrusion detection frameworks can boost detection accuracy and generate human-readable explanations for flagged anomalies, thereby enhancing awareness and reducing false alarms for security analysts [6].

Prior work shows growing interest in applying LLMs to network intrusion detection, with studies highlighting benefits such as contextual reasoning, rapid adaptation to unseen threats, and support for analyst workflows [7]–[11], [11]–[13]. Empirical studies demonstrate promising accuracy from fine-tuned or in-context LLMs (e.g., GPT-4 with few-shot prompts), but these evaluations often occur in generic enterprise or IoT scenarios [14], [15], with limited focus on SDN. While early SDN efforts integrate Transformer-based models for DDoS detection [16] or propose fine-tuned BERT-style architectures [17], [18], these works emphasize accuracy and overlook efficiency and explainability. Efficiency-oriented LLM-based network intrusion detection studies compress Transformer backbones using INT8 or half-precision quantization to meet edge constraints [19], [20], while surveys suggest more aggressive 4-bit strategies (e.g., QLoRA, HQQ) that remain under-explored in operational network intrusion detection pipelines [21]. In terms of explainability, recent frameworks either prompt LLMs to generate natural language rationales [22] or append post-hoc SHAP/LIME visualizations, but they seldom examine how token-level attributions map to structured flow features or document interpretability failure modes. Moreover, most prior LLM-based network intrusion detection research emphasizes detection metrics and occasionally latency or memory usage [19], [20], [23], but does not report energy footprints essential for sustainable deployment. Despite SDN's unique control architecture and attack surface, few LLM-based studies have evaluated models directly on SDN-specific traffic or benchmarks [24], leaving a gap in deploying efficient and explainable LLMs for SDN.

In this paper, we present a lightweight, explainable LLM-based network intrusion detection for SDN that fine-tunes GPT-Neo, Phi-2, and LLaMA2-7B with 4-bit QLoRA on the InSDN dataset, transforms structured flow features into natural-language prompts for binary (normal vs. attack) classification, and jointly measures accuracy, energy, and explainability to characterize deployability. We take this path because SDN requires detectors that are both transparent (to support operator trust and response) and resource-efficient (to run close to controllers/edges), while prior LLM works rarely target SDN traffic, seldom use aggressive 4-bit quantization, and almost never report end-to-end energy/emissions; moreover, existing XAI reports for LLMs do not analyze how

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

token-level attributions behave on structured flow features. Concretely, we (i) convert the 84 InSDN flow features into concise text prompts, (ii) apply QLoRA with 4-bit quantization via BitsAndBytes and FP16 compute, (iii) train with batch size 16 for one epoch and evaluate on held-out data, (iv) instrument training/inference with CodeCarbon to log CO_2, CPU/GPU power, and energy (kWh), (v) compare against widely used classical baseline models (e.g., XGBoost, RF), and (vi) employ SHAP to attribute predictions while documenting the limitations of token-level explanations for LLMs. In summary, the main contributions of this work are:

- We present an LLM–based network intrusion detection approach for SDN that fine-tunes GPT-Neo, Phi-2, and LLaMA2-7B with 4-bit QLoRA, and converts 84 structured flow features into concise natural-language prompts for binary (normal vs. attack) classification.
- We measure the accuracy–efficiency trade-off by tracking CO_2 emissions, power, and energy use, showing that Phi-2 is the most energy-efficient and LLaMA2-7B has the largest footprint.
- We integrate SHAP to attribute LLM predictions and analyze the limitations of token-level explanations on textified flow inputs, contrasting them with clearer, feature-level attributions from tree-based models.
- We benchmark against baselines (e.g., XGBoost) and provide a reproducible configuration (QLoRA hyperparameters and prompts), establishing a practical baseline for deployable, explainable, and resource-efficient LLM-based intrusion detection for SDN.

The rest of the paper is organized as follows: Section II reviews the recent related works, Section III presents our framework, Section IV reports evaluations, and Section V concludes our research with limitations and future work.

II. Related Work

Advancements in Large Language Models (LLMs) have attracted the attention of researchers in the domain of network anomaly detection [9]–[11], [11]–[13]. This section surveys existing literature on the application of LLMs in network anomaly detection, with particular emphasis on SDN use cases, efficiency optimization, and explainability.

A recent survey by Bovenzi et al. [7] documents several use cases of LLMs in network monitoring and management, concluding that network anomaly detection can benefit from LLMs' contextual reasoning and broader understanding of event context. This contextual awareness enables the detection of anomalies by considering richer semantics, and the adaptability of LLMs allows for quicker adjustment to new or evolving threats. Halvorsen et al. [8] similarly discuss how generative AI can support intrusion detection (e.g., by supplementing training data or aiding model development), highlighting that both the training and detection phases of a network anomaly detection can benefit from these models. Balasubramanian et al. [14] fine-tune GPT-3 models as anomaly detectors on log data, achieving high detection accuracy. Such results hint at the promise of LLMs, but mostly in controlled settings. Zhang et al. [15] explore an in-context learning approach for network anomaly detection, where a pre-trained GPT-4 model is fed a few labeled examples of network traffic (as prompts) without any fine-tuning. Remarkably, with only 10 prompt examples, GPT-4 achieves over 95% accuracy in distinguishing between normal and malicious traffic. These results suggest that large pre-trained models can generalize well from minimal context in network security tasks.

Quantization-based compression has emerged as a key strategy for the deployment of LLM-driven network anomaly detection under resource constraints. Several studies compress large Transformer models (e.g., BERT or GPT-style) to 8-bit or lower precision to reduce memory and latency overhead. For example, Adjewa et al. [19] quantize a BERT-based network anomaly detection system for federated 5G networks, trimming its layers and applying per-channel INT8 quantization to reduce the model size by approximately 92.8% with small accuracy loss. This optimization enables real-time inference on edge devices in collaborative intrusion detection settings. Similarly, Rajapaksha et al. [20] report that post-training half-precision quantization (float16) of their LLM anomaly detection system yields a 78% drop in memory use and 83% faster detection latency on a Raspberry Pi, while preserving over 99% detection rates. Recognizing the efficacy of such techniques, a recent survey highlights advanced methods, such as Half-Quadratic Quantization (HQQ) and 4-bit Quantized Low-Rank Adapters (QLoRA), to compress LLMs for security tasks [21]. Collectively, these works demonstrate that aggressive quantization (e.g., 4-bit or INT8) can drastically reduce LLM inference costs for network anomaly detection without compromising detection fidelity.

LLMs have also been explored for securing SDN environments, given their strength in analyzing network data and identifying complex traffic patterns in real time [25]. Early efforts integrate transformer architectures into SDN intrusion detectors [16], [17]. For example, Wang and Li's DDosTC hybrid model [16] combines an attention-based Transformer with a CNN to detect SDN-targeted DDoS attacks, outperforming prior deep learning methods on the CICDDoS2019 benchmark. More recent work has leveraged fine-tuning of LLMs on SDN data: Ataa et al. [17] build an encoder-only Transformer network intrusion detection system for the SDN controller using the InSDN dataset, achieving 99% detection accuracy comparable to a CNN–LSTM baseline. Swileh et al. [18] further fine-tune BERT on InSDN flow features (transformed into textual "sentences"), enabling the detection of both known and zero-day attacks with high precision and recall results. In addition, researchers have begun to exploit LLMs for explainability [22], [23]. For example, Houssel et al. [22] introduce the eX-NIDS framework, in which an LLM (GPT-4 or LLaMA) is prompted with enriched context (e.g., threat intelligence) to generate human-readable rationales for why an SDN flow is flagged as malicious. Given the computational overhead of large models, recent studies emphasize model optimization (e.g., quantization or lightweight fine-tuning) to meet SDN's real-time and resource constraints [23]. These

studies show LLMs enhance accuracy and explainability in SDN intrusion detection, making them promising.

Limitations & Gaps: Despite encouraging results, prior studies leave several gaps that our work addresses. Most evaluations target generic enterprise or IoT traffic [14], [19] rather than SDN, leaving the suitability of LLMs for SDN controllers and data planes less explored. While a handful of recent works apply Transformer-based models to SDN intrusion detection [16]–[18], these studies primarily focus on maximizing detection accuracy on specific SDN benchmarks (e.g., DDoS attack traffic) and do not examine model efficiency or interpretability. Efficiency efforts commonly adopt INT8 or half-precision post-training quantization [19], [20], [23], while more aggressive 4-bit pathways (e.g., QLoRA) have been noted as promising yet remain largely unrealized in network anomaly detection pipelines [21]. In addition, works using LLMs for explanation typically provide high-level natural language rationales (for example, by prompting an LLM to justify alerts [22]) or attach post hoc SHAP/LIME interpretations to a detector, but rarely investigate how token-level attributions align with structured flow features or analyze the resulting interpretability failure modes. Existing studies emphasize detection accuracy and sometimes latency or memory usage [19], [20], yet they do not report end-to-end energy or emissions footprints. In contrast, our study focuses on SDN and evaluates LLMs on the InSDN dataset [24] using an energy-aware, efficiency-centric methodology: we fine-tune GPT-Neo, Phi-2, and LLaMA2-7B with 4-bit QLoRA; transform structured flow features into natural-language prompts for binary classification; integrate SHAP to attribute predictions (and document its limitations for tokenized text); and record power and CO_2 metrics during training and inference. Our SDN-specific, quantization-driven, explainable, and energy-focused design moves LLM-based network intrusion detection closer to practical deployment.

III. Proposed Methodology

This study proposes an LLM-based framework with explainability for predicting SDN attacks, as shown in Figure 1. We conduct experiments using the benchmark InSDN dataset and design prompt templates that incorporate network traffic features. The LLM is fine-tuned using the QLoRA framework, which combines quantization and LoRA for parameter-efficient fine-tuning. In LoRA, the original weight matrix W is frozen, and a low-rank decomposition is applied such that $W' = W + \Delta W$, where $\Delta W = AB$. Here, $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times k}$ are small trainable matrices with rank $r \ll \min(d, k)$. This design reduces the number of trainable parameters while maintaining model performance, and the quantization step compresses the frozen weights for efficient deployment. The framework is evaluated based on classification accuracy and its environmental impact during both training and deployment. Additionally, we integrate the SHAP (SHapley Additive exPlanations) technique to interpret the LLM model's predictions. For a given prediction $f(x)$, SHAP assigns an importance value ϕ_i to each feature x_i by considering all possible feature combinations: $\phi_i = \sum_{S\subseteq F\setminus\{i\}} \frac{|S|!\,(|F|-|S|-1)!}{|F|!}\left[f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)\right]$ where F is the full set of features, S is a subset of features excluding i, and $f_S(x_S)$ is the model prediction using only the subset S. Intuitively, SHAP measures how the prediction changes when the feature x_i is added to all possible subsets of other features, thereby providing a fair and interpretable attribution of each feature's contribution to the final prediction.

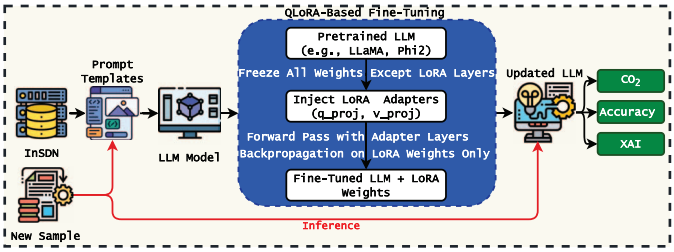

Fig. 1. Overview of Proposed Methodology

A. Dataset Description

This study utilizes the InSDN [24] contains 342,889 records comprising 84 network flow features extracted using sFlow, representing statistical, behavioral, and network-level attributes such as IP addresses, ports, packet size, flow duration, byte count, inter-arrival time, and protocol type. The original dataset includes nine traffic categories (e.g., Probe, DDoS, DoS, Normal, BFA), but to reduce class imbalance and focus on binary classification, all attack types were merged into a single "attack" label, while benign traffic was labeled "normal". We designed the prompt structure in the form *"'Feature Name' is 'Value'"* to balance two competing requirements: (i) preserving the semantics of each structured flow feature for downstream interpretability, and (ii) enabling the LLM tokenizer to process features in a consistent, context-rich format. This format avoids arbitrary numeric-only inputs, which can lead to poor tokenization and unstable training, while still keeping prompts concise enough for efficient fine-tuning. Although more complex natural language descriptions could be used, we selected this lightweight format to ensure reproducibility and token efficiency. We convert our data into a simple prompt, which we pass to models as shown below:

Sample Prompt

Flow ID is 192.168.20.133-74.125.193.138-32768-443-6, Src IP is 192.168.20.133, Src Port is 32768, ..., Active Max is 0.0, Active Min is 0.0, Idle Mean is 0.0, Idle Max is 0.0, Idle Min is 0.0, Label Attack

B. Deployed Models Setup

All experiments were run on a Dell workstation with an Intel Xeon w9-3495X CPU, an NVIDIA RTX 6000 Ada GPU, Windows 11, Python 3.12.10, and CUDA 11.8. The environment included PyTorch 2.2.0, Transformers 4.38.2, Datasets

2.16.1, PEFT 0.7.1, TRL 0.7.10, and supporting libraries (Pandas, Scikit-learn, Matplotlib, TQDM, CodeCarbon) for model training, quantization (LoRA/QLoRA), and energy tracking.

1) Deployed LLMs Architecture: We deploy three large language model architectures, GPT-Neo, Phi-2, and LLaMA 2-7B, based on their lightweight design and significant impact in the literature. We select these models for their efficient architectures, which balance performance with computational requirements, making them suitable for a variety of tasks. Their proven effectiveness in academic and industry applications further motivates their use in this study.

⇒**GPT-Neo 2.7B**[1] is built on the Transformer architecture with 32 layers, each containing 2560 hidden units. It uses 32 attention heads in each transformer block, which allows the model to focus on different parts of the input sequence in parallel. The model's feedforward network has 10,240 units, enabling richer transformations of the hidden states. The activation function used is GELU (Gaussian Error Linear Unit), which provides smoother activations compared to ReLU.

⇒**Phi-2**[2]**,** developed by Google DeepMind, is designed to scale efficiently with an architecture featuring around 64 or more layers. Each transformer block in Phi-2 has up to 14,000 hidden units, allowing the model to capture highly complex patterns in the data. The number of attention heads in Phi-2 is considerably larger, often exceeding 50, giving the model the ability to focus on multiple aspects of the input sequence simultaneously. The feedforward network in Phi-2 is also significantly larger, further enhancing the model's capacity to handle complex transformations.

⇒**LLaMA 2-7B**[3]**,** released by Meta, features 32 layers, similar to GPT-Neo, but each layer has 4096 hidden units, which allows for richer representations of the data. The model uses 32 attention heads, allowing it to effectively capture multiple relationships within the input sequence. The feedforward network in LLaMA 2-7B has 16,384 units, providing a higher capacity for transforming and learning from complex data. Like GPT-Neo, LLaMA 2 uses GELU activation, promoting smoother learning dynamics.

2) Fine-Tuning Of LLMs: This study fine-tunes models using a QLoRA (Quantized Low-Rank Adapter) approach for binary classification on a structured SDN intrusion detection dataset. The pipeline begins by preprocessing the dataset containing up to 350,000 samples. Each sample's features are transformed into natural language prompts like feature_name is value, concatenated to form the input text. The data is split into training and validation sets, and labels are binarized into normal or attack. GPT_NEO, Phi-2, Llama2-7b models are quantized to 4-bit precision using BitsAndBytes for memory-efficient training. QLoRA is applied using LoRAConfig targeting the q_proj and v_proj attention layers[4]. The HuggingFace Trainer API orchestrates the training with a batch size of 16, one epoch, FP16 precision, and logging enabled. After training, the model is evaluated, and predictions are saved. SHAP values are used for interpretability, and *CodeCarbon* tracks emissions during training to quantify the environmental impact. This setup is optimized for low-resource fine-tuning on large language models, ensuring reproducibility and efficiency through its configured settings. Table I shows the parameter settings for fine-tuning the model.

[1]https://huggingface.co/docs/transformers/en/model_doc/gpt_neo

[2]https://huggingface.co/microsoft/phi-2

[3]https://huggingface.co/meta-llama/Llama-2-7b

[4]The q_proj and v_proj refer to the linear projection layers for generating the queries *(q)* and values *(v)* used in the attention computation.

TABLE I
HYPERPARAMETER SETTINGS FOR QLORA-BASED FINE-TUNING

Parameter	Value
Models Names	GPT_NEO, Phi-2, Llama2-7b
LoRA Rank (r)	8
LoRA Alpha	32
LoRA Dropout	0.05
Quantization	4-bit (NF4)
Compute Dtype	float16
Double Quantization	True
Max Input Length	512 tokens
Training Epochs	1
Batch Size	16
Learning Rate	Default
Gradient Checkpointing	Enabled
Loss Function	CrossEntropyLoss
Optimizer	AdamW (via Trainer)
Mixed Precision	FP16

IV. EVALUATION & RESULT DISCUSSION

Table II shows the performance evaluation of three language models, GPT_NEO, Llama2 7b, and Phi-2, across three different data sizes (25k, 100k, 350k). Metrics reported include accuracy (Acc), precision (Pre), recall (Rec), F1 (F1), error rate (ER), correct predictions (CP), and wrong predictions (WP). Here, CP represents the total number of correct predictions, while WP indicates the number of flows misclassified. All models exhibit improved performance with increased training data. Notably, Llama2 7b and Phi-2 achieve high scores of 1.00 on all metrics for the 350k dataset. Even at 25k and 100k levels, their error rates remain extremely low. GPT_NEO, while slightly behind at smaller data scales, also reaches high accuracy at 350k. These results show that all three models are highly capable, with Llama2 7b and Phi-2 showing consistent robustness even with limited data.

TABLE II
PERFORMANCE COMPARISON OF LLM MODELS

Model	Acc	Pre	Rec	F1	ER	CP	WP
GPT_NEO_25000	0.96	0.94	0.86	0.91	0.04	4821	179
GPT_NEO_100000	0.99	0.99	0.99	0.99	0.00	19997	3
GPT_NEO_350000	1.00	1.00	1.00	1.00	0.00	55022	0
Llama2 7b_25000	1.00	1.00	1.00	1.00	0.00	5000	0
Llama2 7b_100000	0.99	1.00	0.99	0.99	0.00	8018	1
Llama2 7b_350000	0.99	0.99	1.00	0.99	0.00	54738	1
Phi-2_25000	0.99	0.99	0.99	0.99	0.00	4997	3
Phi-2_100000	0.99	1.00	0.99	0.99	0.00	19994	6
Phi-2_350000	1.00	1.00	1.00	1.00	0.00	55022	0

Table III presents the performance of the widely used baseline ML models. Among our selected models, XGB and GBM demonstrate the highest performance, with high accuracy scores across all metrics. NB shows the lowest performance, particularly in AUC of 0.76 and recall of 0.76,

indicating limitations in accurately identifying positive cases. LR, while slightly behind the ensemble models, still performs well with an accuracy of 0.93 and an AUC of 0.96. However, when comparing classical models trained on approximately 350k samples with LLMs trained on 100k and 350k samples, the former show slightly lower accuracy. Nonetheless, their computational cost is significantly lower than that of LLMs.

TABLE III
COMPARISON OF BASELINE ML MODELS

Model	Acc	AUC	Pre	Rec	F1	CP	WP
LR	0.93	0.96	0.91	0.88	0.89	64219	4559
DT	0.99	0.99	0.99	0.99	0.99	68764	14
RF	0.99	0.99	0.99	0.99	0.99	68769	9
NB	0.89	0.76	0.91	0.76	0.81	61840	6938
GBM	0.99	0.99	0.99	0.99	0.99	68746	32
XGB	0.99	0.99	0.99	0.99	0.99	68773	5

Table IV presents a comparative summary of carbon emissions produced by three language models, GPT_NEO, Llama2-7b, and Phi-2, at different data usage levels (25,000, 100,000, and 350,000 samples). Across all dataset sizes, Llama2 7b is the most computationally and environmentally expensive model. At just 25,000 samples, it emits 0.304 kg CO_2eq, which is nearly 19 times higher than GPT_NEO, which is 0.016 kg and 77% higher than Phi-2, which is only 0.173 kg, while drawing up to 285.21 W of GPU power. Its GPU energy usage peaks at 1.205 kWh when trained on 350,000 samples. GPT_NEO shows moderate resource usage and emissions. While its emissions rise with data size, from 0.016 kg (25k) to 0.173 kg (350k), its GPU energy consumption remains relatively flat (around 0.57 kWh), and GPU power ranges between 287.13 W and 294.05 W. Phi2, in contrast, is the most efficient model, showing consistent emissions of 0.173 kg CO_2eq across all dataset sizes, regardless of scale. Its GPU energy usage stays exceptionally low, from 0.030 kWh at 25k to just 0.326 kWh at 350k, with a modest GPU power draw ranging from 254.43 W to 265.57 W. According to the statistics, Llama2-7B offers top performance with high energy cost, Phi-2 is most efficient, and GPT-NEO provides a balanced trade-off.

TABLE IV
MODEL EMISSIONS AND SYSTEM CONFIGURATION

Model	Data	EM	ER	CPUE	GPUE	GPUP
GPT_NEO	25000	0.016	3.9E-05	0.084	0.582	294.045
GPT_NEO	100000	0.063	4.1E-05	0.082	0.571	293.698
GPT_NEO	350000	0.173	4.1E-05	0.082	0.567	287.130
Llama2 7b	25000	0.304	4.3E-05	0.083	0.575	285.206
Llama2 7b	100000	0.298	4.3E-05	0.083	0.574	276.477
Llama2 7b	350000	0.296	4.3E-05	0.174	1.205	275.243
Phi2	25000	0.173	4.1E-05	0.005	0.030	265.566
Phi2	100000	0.173	4.1E-05	0.018	0.120	254.435
Phi2	350000	0.173	4.1E-05	0.050	0.326	264.617

EM – CO_2 (Carbon Dioxide Emissions); **GPUE** – GPU Energy (in Kilowatt-hours); **CPUE)** – CPU Energy (in Kilowatt-hours); **GPUP** – GPU Power (in Watts); **Data (Rows)** – Number Of Rows;

SHAP Interpretation on LLMs: Figure 2 shows the SHAP output for Phi-2, which performs well in terms of accuracy and efficiency. However, the SHAP output in Figure 2 does not clearly indicate which features contribute most to the prediction. When SHAP is applied to LLMs, the resulting explanations often appear fragmented and difficult to interpret due to token-level attributions on subword units. This leads to outputs containing unintelligible token combinations, e.g., `0 + + id + le + min + is + 0 + .`, which lack semantic clarity. This occurs because SHAP is inherently designed for structured input features rather than dense textual embeddings.

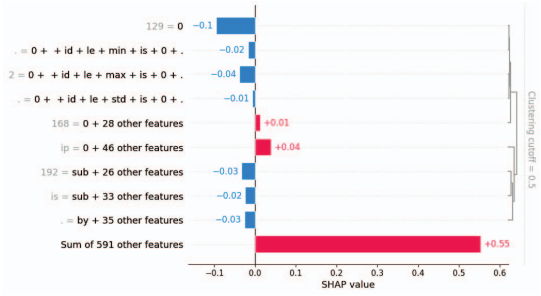

Fig. 2. SHAP For Phi-2 Model

Figure 3 shows the SHAP plot for the XGBoost model, highlighting feature importance and their impact on the model's predictions. Features such as `Pkt Len Max`, `Init Bwd Win Byts`, and `Bwd Header Len` exhibit the highest influence on the model's output. Each dot represents a SHAP value for a specific feature in a given instance, with color indicating the feature value (red for high, blue for low). The spread and concentration of SHAP values provide insights into how variations in feature values affect predictions, supporting interpretability of the model's decision-making process in a structured feature space.

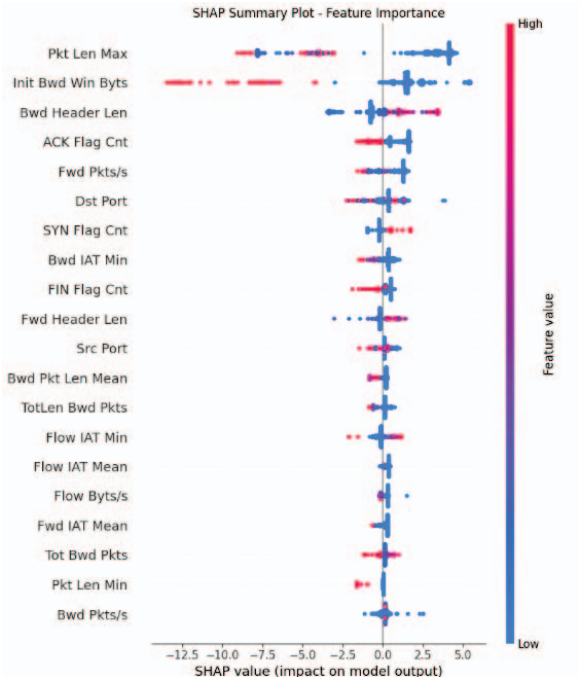

Fig. 3. SHAP For Baseline XGBoost Model

SHAP Limitation: The token-level granularity of SHAP explanations results in poor interpretability for LLMs. It struggles to relate importance scores back to coherent input segments, thereby limiting its usefulness for understanding model behavior without further post-processing or adaptation.

V. Conclusion & Future Work

This paper investigates lightweight, explainable LLMs for SDN intrusion detection. We fine-tune GPT-Neo, Phi-2, and LLaMA2-7B with 4-bit QLoRA on InSDN, transform 84 flow features into compact prompts for binary classification, and evaluate accuracy together with power, energy, and CO_2 footprints. All three LLMs achieve strong performance across data regimes; at 350k samples, LLaMA2-7B and Phi-2 reach *1.00* on Acc/Pre/Rec/F1, and GPT-Neo also attains *1.00*, demonstrating robust detection under increasing data scales. Compared with classical baselines, LLMs outperform them at higher data scales, indicating competitive accuracy with added explanatory capability. Energy profiling reveals a clear trade-off: LLaMA2-7B delivers top accuracy but incurs the highest environmental cost (e.g., 0.304 kg CO_2 at 25k; 1.205 kWh GPU energy at 350k), whereas Phi-2 is consistently the most efficient (0.173 kg CO_2; 0.030–0.326 kWh GPU energy), and GPT-Neo provides a balanced middle ground (0.016–0.173 kg CO_2; $\approx$0.57 kWh). Further, we applied SHAP with LLMs, and we concluded that SHAP often produces fragmented and unintelligible explanations due to token-level attributions on subword units (e.g., `0 + + id + le + min + is + 0 + .`). This arises because SHAP is designed for structured input features rather than dense textual embeddings.

In future work, we will develop feature-aligned XAI methods to project LLM predictions onto original SDN flow features, making outputs more interpretable and operator-friendly, thus addressing current token-level interpretability limits and improving trust. At the same time, we will extend the real-time deployment framework to operational SDNs, aiming to minimize inference latency, reduce energy costs, and ensure compatibility with controller security tasks.

References

[1] A. V. Turukmane and R. Devendiran, "M-multisvm: An efficient feature selection assisted network intrusion detection system using machine learning," *Computers & Security*, vol. 137, p. 103587, 2024.

[2] A. H. Abdi, L. Audah, A. Salh, M. A. Alhartomi, H. Rasheed, S. Ahmed, and A. Tahir, "Security control and data planes of sdn: A comprehensive review of traditional, ai, and mtd approaches to security solutions," *IEEE Access*, vol. 12, pp. 69941–69980, 2024.

[3] V. Z. Mohale and I. C. Obagbuwa, "A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity," *Frontiers in Artificial Intelligence*, vol. 8, p. 1526221, 2025.

[4] S. Elouardi, A. Motii, M. Jouhari, A. N. H. Amadou, and M. Hedabou, "A survey on hybrid-cnn and llms for intrusion detection systems: Recent iot datasets," *IEEE Access*, 2024.

[5] A. Angi, A. Sacco, and G. Marchetto, "Llnet: An intent-driven approach to instructing softwarized network devices using a small language model," *IEEE Transactions on Network and Service Management*, 2025.

[6] B. Karunanayake, I. Khalil, X. Yi, and K.-Y. Lam, "Toward llm-driven adaptive policy orchestration for host-based intrusion detection systems in iot environments," *IEEE Network*, 2025.

[7] G. Bovenzi, F. Cerasuolo, D. Ciuonzo, D. Di Monda, I. Guarino, A. Montieri, V. Persico, and A. Pescapé, "Mapping the landscape of generative ai in network monitoring and management," *IEEE Transactions on Network and Service Management*, 2025.

[8] J. Halvorsen, C. Izurieta, H. Cai, and A. Gebremedhin, "Applying generative machine learning to intrusion detection: A systematic mapping study and review," *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–33, 2024.

[9] O. G. Lira, A. Marroquin, and M. A. To, "Harnessing the advanced capabilities of llm for adaptive intrusion detection systems," in *International Conference on Advanced Information Networking and Applications*, pp. 453–464, Springer, 2024.

[10] L. Coppolino, S. D'Antonio, G. Mazzeo, and F. Uccello, "The good, the bad, and the algorithm: The impact of generative ai on cybersecurity," *Neurocomputing*, vol. 623, p. 129406, 2025.

[11] L. Gutiérrez-Galeano, J.-J. Domínguez-Jiménez, J. Schäfer, and I. Medina-Bulo, "Llm-based cyberattack detection using network flow statistics," *Applied Sciences*, vol. 15, no. 12, p. 6529, 2025.

[12] X. Zhang, H. Meng, Q. Li, Y. Tan, and L. Zhang, "Large language models powered malicious traffic detection: Architecture, opportunities and case study," *IEEE Network*, 2025.

[13] F. Adjewa, M. Esseghir, L. Merghem-Boulahia, and C. Kacfah, "Llm-based continuous intrusion detection framework for next-gen networks," in *2025 IWCMC*, pp. 1198–1203, IEEE, 2025.

[14] P. Balasubramanian, J. Seby, and P. Kostakos, "Transformer-based llms in cybersecurity: An in-depth study on log anomaly detection and conversational defense mechanisms," in *2023 IEEE International Conference on Big Data (BigData)*, pp. 3590–3599, 2023.

[15] H. Zhang, A. B. Sediq, A. Afana, and M. Erol-Kantarci, "Large language models in wireless application design: In-context learning-enhanced automatic network intrusion detection," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*, pp. 2479–2484, IEEE, 2024.

[16] H. Wang and W. Li, "Ddostc: A transformer-based network attack detection hybrid mechanism in sdn," *Sensors*, vol. 21, no. 15, p. 5047, 2021.

[17] M. S. Ataa, E. E. Sanad, and R. A. El-Khoribi, "Intrusion detection in software defined network using deep learning approaches," *Scientific Reports*, vol. 14, no. 1, p. 29159, 2024.

[18] M. N. Swileh and S. Zhang, "Unseen attack detection in software-defined networking using a bert-based large language model," *AI*, vol. 6, no. 7, p. 154, 2025.

[19] F. Adjewa, M. Esseghir, and L. Merghem-Boulahia, "Efficient federated intrusion detection in 5 g ecosystem using optimized bert-based model," in *2024 20th WiMob*, pp. 62–67, 2024.

[20] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, and G. Madzudzo, "Improving in-vehicle networks intrusion detection using on-device transfer learning," in *VehicleSec*, vol. 10, 2023.

[21] M. A. Ferrag, F. Alwahedi, A. Battah, B. Cherif, A. Mechri, N. Tihanyi, T. Bisztray, and M. Debbah, "Generative ai in cybersecurity: A comprehensive review of llm applications and vulnerabilities," *Internet of Things and Cyber-Physical Systems*, 2025.

[22] P. R. Houssel, S. Layeghy, P. Singh, and M. Portmann, "ex-nids: A framework for explainable network intrusion detection leveraging large language models," *arXiv preprint arXiv:2507.16241*, 2025.

[23] S. Yang, X. Zheng, X. Zhang, J. Xu, J. Li, D. Xie, W. Long, and E. C. Ngai, "Large language models for network intrusion detection systems: Foundations, implementations, and future directions," *arXiv preprint arXiv:2507.04752*, 2025.

[24] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "Insdn: A novel sdn intrusion dataset," *IEEE access*, vol. 8, pp. 165263–165284, 2020.

[25] H. Kheddar, "Transformers and large language models for efficient intrusion detection systems: A comprehensive survey," *Information Fusion*, p. 103347, 2025.